Amendments to the claims,
Listing of all claims pursuant to 37 CFR 1.121(c)

This listing of claims will replace all prior versions, and listings, of claims in the application:

What is claimed is:

1. (Previously presented) A method for extended memory support in a database system having a primary cache for storing database pages, the method comprising:

using a memory mapped file, creating a secondary cache in system memory available to the database system;

mapping a virtual address range to at least a portion of the secondary cache;

when the primary cache is full, replacing database pages from the primary cache using the secondary cache;

in response to a request for a particular database page, searching for the database particular page in the secondary cache if the particular database page is not found in the primary cache;

if the particular database page is found in the secondary cache, determining a virtual address in the secondary cache where the particular database page resides based on the mapping; and

swapping the particular database page found in the secondary cache with a database page in the primary cache, so as to replace a database page in the primary cache with the particular database page from the secondary cache.

2. (Original) The method of claim 1, wherein said creating step includes creating the secondary cache using a shared memory file system.

3. (Original) The method of claim 2, wherein the shared memory file system is available as part of an operating system on a computer platform on which the database system is running.

4. (Original) The method of claim 3, wherein the operating system comprises a

Linux operating system.

5. (Original) The method of claim 1, wherein said mapping step includes using a memory mapped file function.

6. (Original) The method of claim 5, wherein the memory mapped file function is available as part of an operating system on a computer platform on which the database system is running.

7. (Original) The method of claim 1, wherein said creating step includes creating the secondary cache on external memory available to the database system.

8. (Previously presented) The method of claim 1, wherein said swapping step includes consulting a least recently used (LRU) list maintained for the primary cache to determine the database page to be moved to the secondary cache.

9. (Previously presented) The method of claim 8, wherein said swapping step further comprises copying the database page to be moved to the secondary cache to a temporary buffer.

10. (Previously presented) The method of claim 9, wherein said swapping step further comprises moving the particular database page from the secondary cache to address of the database page in the primary cache to be moved to the secondary cache.

11. (Previously presented) The method of claim 9, wherein said swapping step further comprises moving the database page from the temporary buffer to the secondary cache.

12. (Previously presented) The method of claim 11, wherein further comprising:

adding the replaced database page to a most recently used end of a most recently used/least recently used (MRU/LRU) list maintained for the secondary cache.

13. (Previously presented) The method of claim 1, wherein said replacing step includes maintaining a least recently used (LRU) list for the primary cache and selecting the database page to be moved to the secondary cache based on said LRU list.

14. (Previously presented) The method of claim 1, further comprising:

providing a washing mechanism in the secondary cache for writing database pages in the secondary cache to disk.

15. (Previously presented) The method of claim 14, wherein a database page is written from the secondary cache to disk in response to copying a database page from disk to the primary cache.

16. (Previously presented) The method of claim 15, wherein the database page written from the secondary cache to disk is selected, based at least in part, on a most recently used/least recently used (MRU/LRU) list maintained for the secondary cache.

17. (Previously presented) The method of claim 1, wherein said replacing step includes determining database pages to be maintained in the secondary cache.

18. (Previously presented) The method of claim 17, wherein said determining step includes determining database pages to be maintained in the secondary cache based, at least in part, on workload of the database system.

19. (Previously presented) The method of claim 1, wherein said replacing step includes substeps of:

moving a database page from the primary cache to the secondary cache; and

reading a database page into the primary cache from disk.

20. (Previously presented) The method of claim 19, wherein said substep of moving a database page from the primary cache includes selecting a database page from

the primary cache based on a least recently used (LRU) list maintained for the primary cache.

21. (Previously presented) The method of claim 19, wherein said substep of moving a database page from the primary cache includes selecting a location for the database page in the secondary cache based on a most recently used/least recently used (MRU/LRU) list maintained for the secondary cache.

22. (Previously presented) The method of claim 1, further comprising:

storing on a computer-readable medium processor-executable instructions for performing the method of claim 1.

23. (Previously presented) The method of claim 1, further comprising:

downloading a set of processor-executable instructions for performing the method of claim 1.

24. (Previously presented) A database system providing extended memory support, the system comprising:

a primary cache for maintaining data pages used by the database system in addressable memory available to the database system;

a secondary cache, created in system memory using a memory mapped file, for maintaining data pages replaced from the primary cache in extended memory available to the database system;

a search module for receiving a request from a user for a particular data page and determining whether the particular data page is in secondary cache if the particular data page is not in the primary cache; and

a module for replacing a data page in the primary cache with the particular data page from the secondary cache if the particular data page is found in the secondary cache.

25. (Original) The system of claim 24, wherein the secondary cache is implemented using a shared memory file system.

26. (Original) The system of claim 25, wherein the shared memory file system is available as part of an operating system on a computer platform on which the database system is running.

27. (Original) The system of claim 26, wherein the operating system comprises a Linux operating system.

28. (Original) The system of claim 24, wherein the secondary cache is mapped to the extended memory using a memory mapped file function.

29. (Original) The system of claim 28, wherein the memory mapped file function is available as part of an operating system on a computer platform on which the database system is running.

30. (Original) The system of claim 24, wherein the secondary cache is created on external memory available to the database system.

31. (Original) The system of claim 24, wherein the primary cache includes a least recently used (LRU) list for determining data pages to be moved to the secondary cache.

32. (Original) The system of claim 31, wherein the module for replacing consults the LRU list for selecting the data page to be moved to the secondary cache.

33. (Original) The system of claim 24, wherein the module for replacing copies the data page to be moved to the secondary cache to a temporary buffer.

34. (Original) The system of claim 33, wherein the module for replacing moves the particular data page from the secondary cache to address of the data page in the primary cache to be moved to the secondary cache.

35. (Original) The system of claim 33, wherein the module for replacing moves the data page from the temporary buffer to the secondary cache.

36. (Original) The system of claim 35, wherein the secondary cache includes a most recently used/least recently used (MRU/LRU) list and the module for replacing adds the data page moved to the secondary cache to the most recently used end of said MRU/LRU list.

37. (Original) The system of claim 24, further comprising:

a washing mechanism in the secondary cache for writing data pages in the secondary cache to disk.

38. (Original) The system of claim 37, wherein the washing mechanism writes a data page in the secondary cache to disk in response to copying a data page from disk to the primary cache.

39. (Original) The system of claim 38, wherein the washing mechanism selects the data page from the secondary cache based, at least in part, on a most recently used/least recently used (MRU/LRU) list maintained for the secondary cache.

40. (Original) The system of claim 24, wherein the module for replacing determines data pages to be maintained in the secondary cache.

41. (Original) The system of claim 24, further comprising:

a module for reading a page into the primary cache from disk.

42. (Original) The system of claim 41, wherein the module for reading selects a data page from the primary cache to be moved to the secondary cache if the primary cache is full.

43. (Original) The system of claim 42, wherein the module for reading selects the

data page based on a least recently used (LRU) list maintained for the primary cache.

44. (Original) The system of claim 42, wherein the module for reading selects a location in the secondary cache for the data page to be moved from the primary cache.

45. (Original) The system of claim 44, wherein the module for reading selects the location in the secondary cache based on a most recently used/least recently used (MRU/LRU) list maintained for the secondary cache.